

MA-LPSC 21-22

[Accueil](#) / [Mes cours](#) / [MA-LPSC 21-22](#) / 8 mar - Labo / PRESENTIEL / [TP1 Vivado](#)

TP1 Vivado

Travail pratique n°1

L'objectif est de vous permettre de découvrir l'environnement Vivado en un seul travail pratique. Mais également d'évaluer vos connaissances en VHDL.

Il faut dans un premier temps récupérer les fichiers spécifiques à la carte Nexys Video depuis le Moodle MSE et les copier dans le dossier d'installation de Vivado. Il s'agit du dossier **nexys_video**.

...

```
# cp -R <DOSSIER_DE_TELECHARGEMENT>/nexys_video/  
<DOSSIER_RACINE_INSTALLATION_VIVADO>/Xilinx/Vivado/2021.2/data/boards/board_files/  
...
```

Objectif du TP1

L'objectif est de créer une petite mémoire et d'écrire et de lire des données.

Pour cela, on utilisera deux modes. le mode READ et le mode WRITE. Le mode devra être sélectionné à partir d'un bouton virtuel.

En mode WRITE, le comportement (séquentiel) sera le suivant:

1. On sélectionne une adresse (8 bits) avec les switches.
2. On valide l'adresse avec le bouton (physique) BtnD.
3. On sélectionne la donnée (8 bits) à écrire avec les switches.
4. On valide l'écriture de la donnée à l'adresse mémoire voulue avec le bouton BtnD.
5. Dans ce mode, les LEDs affichent successivement l'adresse et la donnée que l'on est en train de saisir avec les switches.

On voit facilement que vous aurez besoin de machines d'états.

En mode READ, le comportement sera le suivant:

1. On sélectionne une adresse (8 bits) avec les switches.
2. Les LEDs affichent alors la donnée contenue à l'adresse mémoire choisie.

On aura probablement besoin quelque part dans le code d'un multiplexeur.

Vivado fourni également des boutons et des leds virtuels. Ceux-ci peuvent être utilisés en instanciant et configurant une IP **VIO** depuis l'**IP Catalog**. Ces boutons et ces leds virtuelles sont utilisables depuis le Hardware Manager.

Lors du TP1, on utilise le bouton central (**BtnC**) comme signal de reset, le bouton bas (**BtnD**) comme bouton de validation d'écriture dans la mémoire, les **8 leds** pour afficher le contenu de la mémoire et les **8 switches** comme selecteur d'adresse et de donnée.

Ces composants sont déjà décommentés dans le fichier de contraintes.

Il faut que les noms de vos ports I/O correspondent aux noms qui se trouve dans le fichier de contraintes.

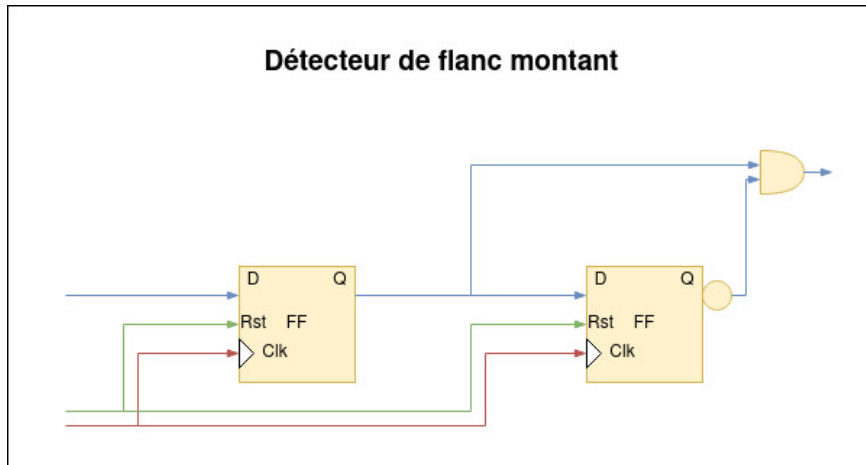
La création du projet

1. Créez un nouveau projet en choisissant la carte **Nexys Video**.
2. Créez un fichier VHDL **top level**.

3. Complétez les ports IO du fichier VHDL **top level** selon les composants que l'on va utiliser sur la carte (voir fichier de contrainte).
4. Créez un fichier de contraintes vide où l'on mettra les signaux que l'on souhaite debugger.
5. Créez un fichier **package** qui contiendra les constantes utiles et les nouveaux type VHDL que vous allez définir.
6. Utilisez ce package dans les fichiers VHDL où il sera nécessaire.
7. Créez un nouveau composant **memory** en inférant une mémoire à partir d'un code VHDL.
8. Instanciez ce composant dans le fichier **top level**.
9. Créez un nouveau composant **edge_detector** (détecteur de flanc montant) en utilisant une des primitives des bascules disponibles dans les slices. La primitive se nomme **FDRE**. Ce composant est utile pour détecter la pression d'un bouton physique.
10. Instanciez ce composant dans le fichier **top level**.
11. Ajoutez une **IP VIO** et configurez la pour avoir un seul bouton virtuel (sur 1 bit).
12. Implémentez votre architecture et vos machines d'états dans le fichier **top_level**.

P.S. N'oubliez pas de lire la page Moodle MSE sur les bonnes pratiques en VHDL.

Shéma d'un détecteur de flanc



Modifié le: mardi 8 mars 2022, 10:14

[◀ Cours sur les mémoires](#)

Aller à...

[Fichier de contraintes pour le TP1 ►](#)

Connecté sous le nom « Müller Quentin » (Déconnexion)
MA-LPSC 21-22

Cours
Français (fr)
Deutsch (de)
English (en)
Français (fr)

Résumé de conservation de données
[Obtenir l'app mobile](#)